Configuring RattleRR for Windows Server 2003

A Liveware Publishing, Inc. White Paper
February 24, 2005
Christian A. Strasser
CTO – Liveware Publishing, Inc.

Introduction

Server security has become one of the most important technology issues facing companies today. Years ago adding a web server to your internal network was handled without much consideration for the consequences. But with the prevalence of worms, bots, zombies, viruses that attack specific vulnerabilities of Windows operating systems and services, organizations and IT professionals have rightly become much more cautious about what they expose to the outside world.

When RattleRR was introduced to the market in 2002, Windows NT and IIS 4.0 were still mainstream products, albeit with known vulnerabilities. Both were configured to be quite "open" out of the box. Windows 2000 and IIS 5.0 had been more secure in their initial configuration, but were still generally open enough to allow many applications to run without difficulty.

Over that same period, however, Microsoft had been getting greater pressure to do something about security. The many updates, patches and fixes being issued were beginning to damage Microsoft's credibility. Their philosophy of providing a system with most of the various services turned on was under heavy criticism as well.

Into this environment, Microsoft introduced Windows 2003 Server along with IIS 6.0. Both were touted as the most secure operating system and web server introduced by them to date. But by turning off many services and by implementing a more extensive set of security and account restrictions, they created problems for many applications that had been developed to rely on the more open configurations.

RattleRR was one such program. It ran fine on NT, W2K and XP, but was totally shutdown in Windows 2003. Executing a RattleRR Active Server page would bring up the browser control, load the executable into memory and stop. Until the process was killed or timed out, the end-user had no idea what was going on.

Resolution

As with many issues that present as broken or buggy behaviors of programs running under Windows operating systems, the root cause of the RattleRR problems was permissions and rights for the system accounts that RattleRR relies on for its access to the server's resources. The difficulty was in determining the proper configuration over the combination of rights, users / accounts impacted and specific programs being run.

We began by creating two test systems: One running Windows2003 Standard Server and the other running Windows2003 Web Server. Initially, we noted that Windows XP Pro was able to run RattleRR from its web server with no problems. Thus, we decided to compare the various services running and permissions granted to the user to determine what was missing and turn them on. In the end, no significant differences were found in the various services active on the two platforms. However, Windows XP runs IIS 5.1 and Windows 2003 runs IIS 6.0. The two web server programs are very different in their approach to default security.

The next step was to engage Microsoft to assist in determining what differences between the two versions of IIS and Windows might be impacting RattleRR. They were very helpful. The solutions presented here represent the efforts of their support under MSDN (both the Developer Team and the IIS Team). There are three possible configuration changes that can be made to enable RattleRR: 1) Run in IIS 5 compatibility mode, 2) Elevate the default right-set used to run IIS 6 or 3) Create an elevated right set to run only RattleRR. Each will be addressed below. All of these configuration changes need to be made in IIS 6. To open the management console to do this, click on Start / All Programs / Administrative Tools / Internet Information Services (IIS) Manager.

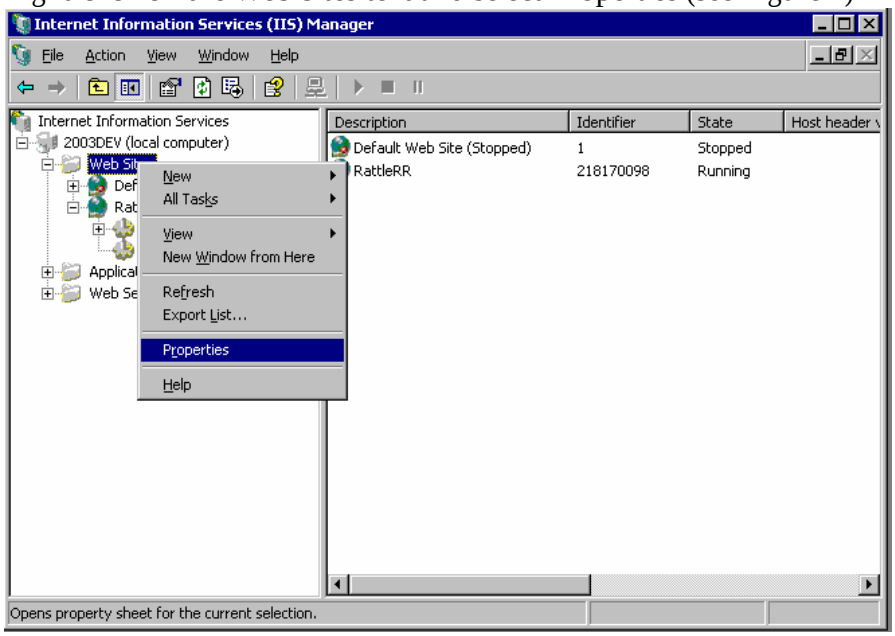
Run in IIS 5 Compatibility Mode

This is the simplest way to reconfigure IIS to run RattleRR. However, it is somewhat more risky to run IIS this way. From the Windows Server System website (http://www.microsoft.com/resources/documentation/WindowsServ/2003/standard/proddocs/en-us/Default.asp?url=/resources/documentation/WindowsServ/2003/standard/proddocs/en-us/arc_modes.asp) , they note that,

> The default identity of applications running in IIS 5.0 isolation mode is LocalSystem, which enables access to and the ability to alter nearly all of the resources on the computer.
>
> Although worker process isolation mode offers increased isolation, reliability, availability, and performance, some applications may have compatibility issues when running in this mode. If you encounter compatibility problems, use IIS 5.0 isolation mode.

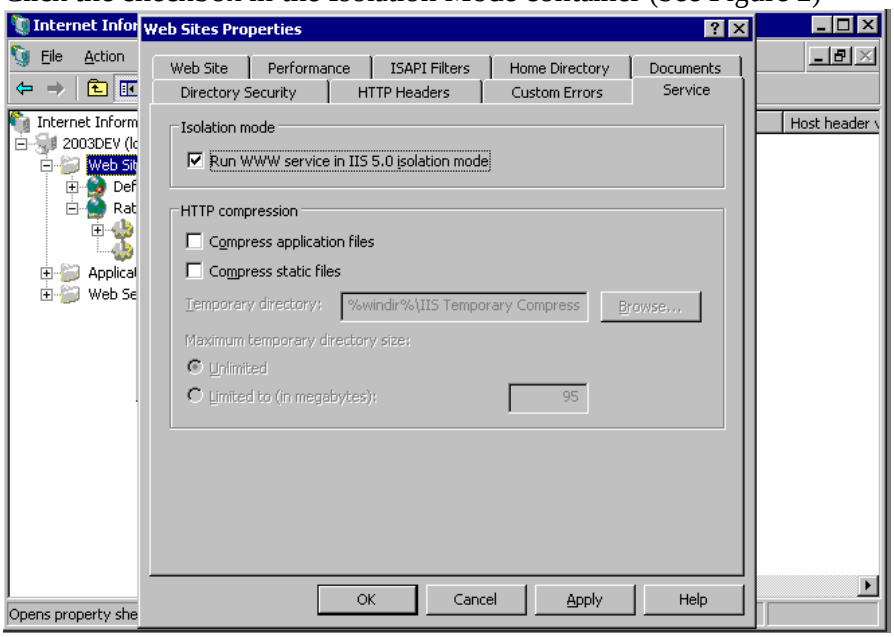
They also proceed to define the circumstances under which you should choose this mode. If you decide to make this change, perform the following actions in IIS 6.0:

1. Expand the tree to display your website
   a. Internet Information Services
   b. Local Computer Name (here it is 2003DEV)
   c. Web Sites
2. Right-click on the Web Sites text and select Properties (see Figure 1)



   a. On the following dialog that appears, click the Service tab

b. Click the checkbox in the Isolation Mode container (See Figure 2)



c. Click Apply (IIS will request a restart of the service); Allow it
d. Click OK
e. Close IIS 6

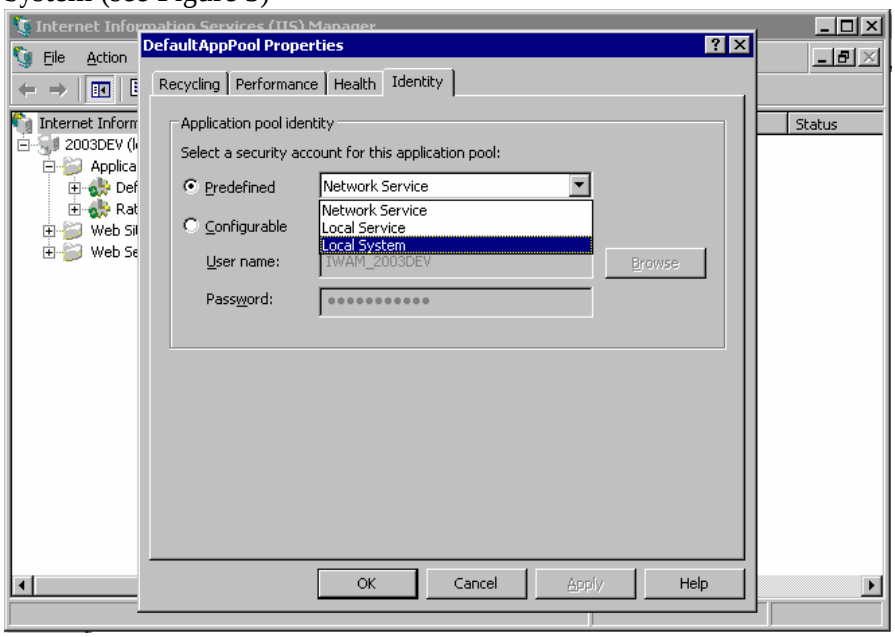
Note that changes made here will apply to all applications running under IIS. Other applications that require IIS 5 isolation mode settings include SourceGear Vault, Soft Artisans FileUp 4.x and Netware File Sharing under Novell.

Elevate Default Right-Set used to run IIS

In principle, making this change is similar to running IIS 6 in IIS 5 isolation mode. However, the configuration changes are more explicitly displayed. The technique here is to change the identity used by the system to perform actions within IIS 6. From the IIS control panel,

1. Expand the tree to display the application pools
   a. Internet Information Services
   b. Local computer name (here it is 2003DEV)
   c. Application Pool
2. Right-click on the Default Application Pool and select Properties
   a. Click on the Identity tab

b. Change the Security Account used in the set of pre-defined identities to Local System (see Figure 3)



c. Click Apply and OK.

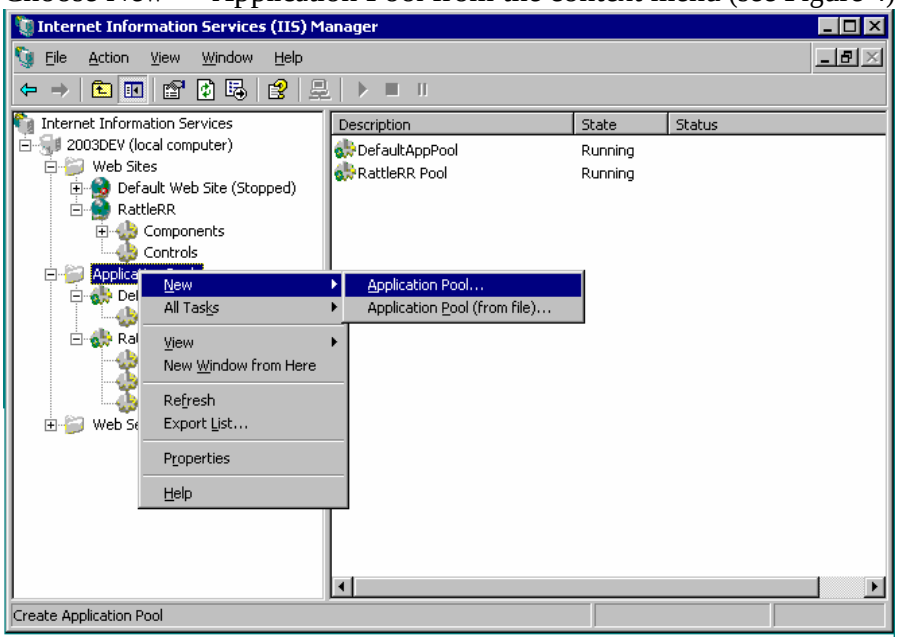
IIS will now use this identity with its security permissions and allowances over all the processes that IIS is used with and that are configured to use the default identity (everything initially). Because of the broad levels of access provided to this account, best practices warn against setting the default to this.

Create Right-Set Applicable only to RattleRR

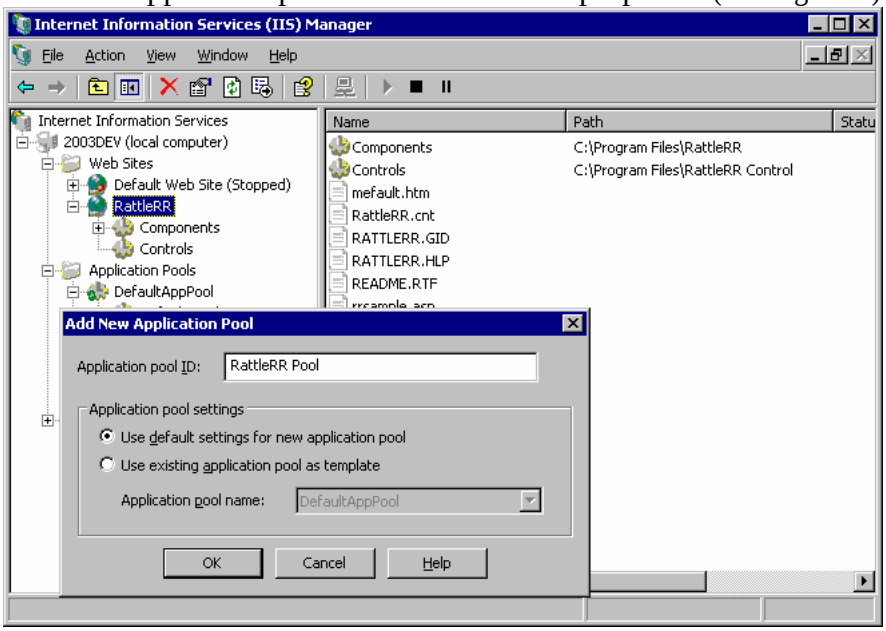
The third technique we'll present to configure RattleRR builds on the one above, but ensures more security, since it will apply only to the RattleRR website and not to all websites / processes / applications running under IIS on the server. Under this scheme, we'll create an application pool based on the default user, but designed to run only for the RattleRR site. Note that this technique presupposes that you've configured RattleRR as a separate website under your overall web server and that you've added the requisite virtual directories. From the IIS control panel:

1. Expand the tree to display the application pools
   a. Internet Information Services
   b. Local computer name (here it is 2003DEV)
   c. Application Pool
2. Create a new Pool
   a. Right-click on Application Pools

b. Choose New -> Application Pool from the context menu (see Figure 4)



c. Name the application pool and set the default properties (see Figure 5)



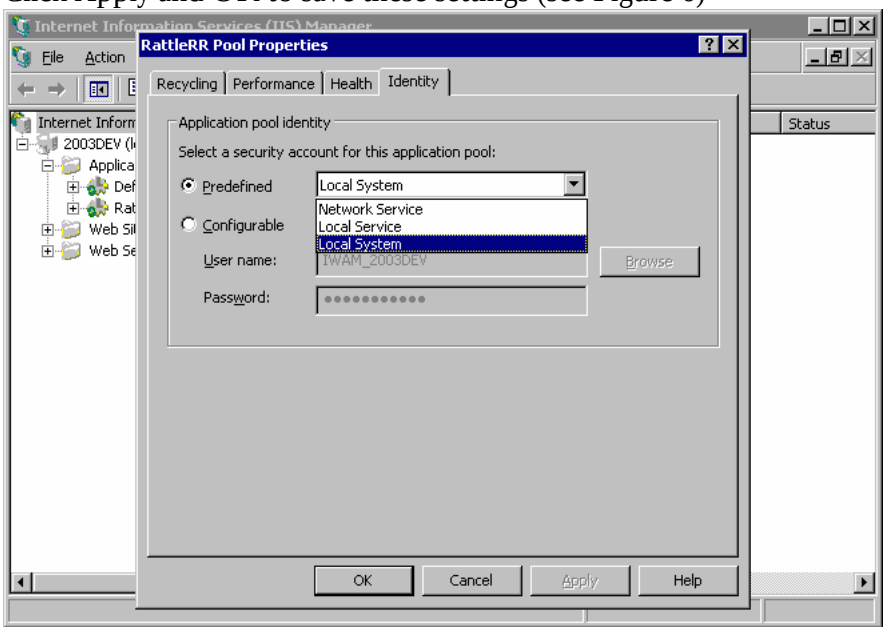
d. Click OK to save this pool.

e. Change the identity associated with this pool to be the Local System

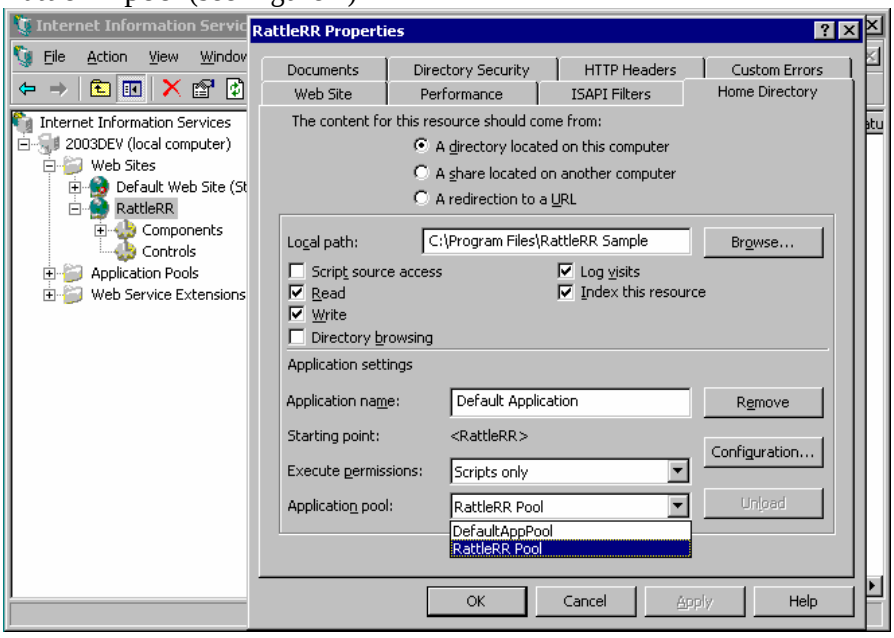
   i. Right-click on the RattleRR Application Pool and select Properties

   ii. Click on the Identity tab and change the pre-defined identity to be Local System from Network Service

iii. Click Apply and OK to save these settings (see Figure 6)



3. Assign the new identity to the RattleRR site
   a. Right-click on the RattleRR website object and select Properties from the context menu
   b. Click on the Home Directory tab and in the Application Pool dropdown, select the RattleRR pool (see Figure 7)



   c. Click on Apply and OK to save the settings.

Conclusion

Liveware is pleased to be able to deliver a versatile tool like RattleRR to our customers with the need for Internet report delivery. We strive to test our tools in all environments to the greatest extent possible. Unfortunately, we aren't always able to verify all combinations. When we learn of incompatibilities or problems, we work as diligently as we can to resolve the issue. This is what we've done here.

We'd also like to stress that making changes to the security settings of any integral Windows application or service is an inherently risky action. It is incumbent on RattleRRs users to take any additional steps necessary to enhance security that opening Windows server rights requires. Liveware is not liable for any server problems or issues that arise from the reconfiguration choices described in this document. Customers perform these modifications at their own risk.